

# Methodology For Viewing Large Strategies Via A Computer Workstation

5

## **BACKGROUND OF THE INVENTION**

### **TECHNICAL FIELD**

10 The invention relates to the organization and viewing of information. More particularly, the invention relates to a methodology for viewing large strategies via a computer workstation.

A strategy is a series of decisions that divide a population into subsets. The first decision divides the population into two or more segments (i.e. partitions).  
15 For each of these segments, a second decision divides the segment into smaller segments. The second decision depends on the choice for the first decision, and the third decision depends on the choice for the second decision, and so on. So strategies can be represented as trees, where each branch of the tree represents  
20 the decisions to make given the choices for earlier decisions. The "root" of the strategy is the first decision.

### **DESCRIPTION OF THE PRIOR ART**

25

Displaying large strategies on a computer display using current methodologies is problematic because the structure of the entire strategy adversely affects the portion of the strategy one wishes to view. Navigation using existing methodologies is problematic because location context of a  
30 segment of the strategy, *i.e.* the decision sequence corresponding to that

segment within the strategy, is not provided, and because navigation takes places by moving around the canvas on which the strategy is displayed, rather than more naturally navigating from segment to segment

## 5 What's wrong with the current tools?

The current crop of strategy display tools have a variety of problems, as follows:

### 10 • **Some tools cannot display all strategies.**

Many of the existing tools cannot display certain kinds of strategies. For example, commercially available strategy viewers can only display strategies that have a single segmentation variable on each level of the strategy.

### 15 • **They do a very poor job of displaying large strategies.**

Historically, strategies have been small. This is certainly true, in large part, because the strategies were constructed manually. One could also  
20 argue that the strategies were kept artificially small due to the fact that large strategies are so difficult to view and interact with using the existing set of tools. In some tools of this type, strategies are automatically constructed and tend to be large and sometimes very large. Even though the strategies are automatically constructed, it is still critical to be able to  
25 view and edit them. Even the most sophisticated of the current crop of tools does a mediocre job of displaying such large strategies.

All these tools employ the same layout technique, *i.e.* they provide enough distance between each pair of segments on the same level of the strategy to fit all of their descendant segments in between them. When displaying a large strategy, the segments closest to the top of the strategy often become very separated, sometimes so distant that it is impossible to see more than one of these segments on the computer screen at the same time. This makes it difficult to edit and understand the conditions at the very top of the strategy, which are the most important conditions in the strategy.

- **Navigation through the strategy is inefficient.**

Navigation techniques in traditional tools, such as scrolling and zooming the canvas upon which the strategy is displayed, are poor approaches to strategy navigation. When the canvas is too large to be displayed on the computer screen in its entirety with an adequate amount of detail, it is all too easy to lose one's place within the strategy and to lose locational context. These tools require the user to know the place on the canvas to which they want navigate. When the strategy becomes even moderately sized, it becomes difficult to remember where on the canvas a certain branch is located. It can require much trial and error before the desired branch is found. Simple navigational goals, such as "Go to the segment containing the segment I am currently looking at" or "View all the descendant segments of the segment I am currently looking at" are difficult to achieve quickly with the traditional tools.

- The condition path is difficult to follow.

The sequence of conditions that must be met in order to reach any segment in the strategy is the condition path. The condition path is a critical element of a segment of the strategy because it answers the questions "How do I get here?" and "What does this segment of the strategy mean?" Despite this critical need, none of the current tools provides a way to view the condition path easily.

No prior approaches exist to solve these problems. It would therefore be advantageous to provide a methodology for viewing large strategies, for example via a computer workstation.

### **SUMMARY OF THE INVENTION**

The invention herein draws from the design principles of the field of information visualization and provides a series of design principles for the efficient display of large strategies, based in part upon the use of fish-eye views. Specific principles include the following:

- Make it very quick and easy to select which portion of the strategy is displayed on screen.
- Do not let the off screen part of strategy affect the display of the on screen part of the strategy.

- Fit as much information on the computer screen as possible, but suppress any information that only causes clutter in the display.

- Always show the condition path.

5

- Rather than navigating by scrolling and zooming a canvas upon which the entire strategy is displayed, navigate by hopping from segment to segment.

- 10
- Provide navigational shortcuts, making it quick and easy to navigate from a segment to a neighboring segment, such as a sibling or descendant segment.

- Provide navigational cues when hopping from one segment to another.

15

- Maintain a consistent top of the strategy-to-leaves orientation.

- Fit the strategy display into a rectangular view.

- 20
- Be capable of rendering all strategies.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

25 Fig. 1 is a display of a conventional strategy view; and

Fig. 2 is a display of a strategy view according to the invention.

## DETAILED DESCRIPTION OF THE INVENTION

The invention herein draws from the design principles of the field of information visualization, advocated by E. Tufte, The Visual Display of Quantitative Information, Graphics Press (February 1992), and specifically makes use of display techniques such as fish-eye views (see, for example, G. Furnas, *Generalized Fisheye Views*, Human Factors in Computing Systems CHI '86 Conference Proceedings, 16-23 (1986).

### **Design Principle 1: Make it very quick and easy to select which portion of the strategy is displayed on screen**

In traditional approaches to displaying a strategy, the whole strategy is rendered on a canvas. A portion of the strategy is selected for display by moving about a rectangular viewing area over the canvas with scroll bars. The portion of the strategy underneath the viewing area is displayed on screen. With this approach, it is common for bits and pieces of multiple strategy branches to appear on screen at once with none of these branches shown in their entirety. Often, individual segments are clipped by the edge of the viewing area so that they are not shown displayed in their entirety. These aspects of the traditional approach make it difficult to choose a portion of the strategy for display.

With the invention herein, the user selects which portion of the strategy to display by choosing a branch to display and how many levels of that branch to

display. A branch is chosen for display simply by clicking the segment at the top of that branch so that this segment becomes the focus segment. Most of the display is devoted to showing just the descendant segments of the focus segment. Segments are never clipped by the display edges. They are either displayed in their entirety or are completely hidden.

When the branch to be displayed is quite large itself, the user can limit the number of levels of the branch being displayed so that he or she can concentrate on the segments near the top of the branch. Those segments beyond the limit are not rendered. The user can specify the limit very quickly, which makes this a fast method for hiding those segments lower on the branch that are not currently of interest.

***Design Principle 2: Do not let the off screen part of a strategy affect the display of the on screen part of the strategy.***

The traditional approach to displaying a large strategy is fundamentally flawed because the entire strategy is first laid out in memory, even though only part of that strategy is viewable at any time. This leads to a phenomenon where the most important part of the strategy, *i.e.* the initial branches nearest to the top of the strategy, cannot be viewed in their entirety with sufficient detail because the off screen portion of the strategy is so bushy. The best one can hope to do is get a sense of those initial branches by scrolling and zooming – a very cumbersome approach.

The approach taken by the invention herein is very different. If a portion of the strategy is not being viewed, it has no effect on the layout of the visible portion of the strategy. This means that the strategy layout is dynamic and adaptable to the current portion of the strategy being viewed. It is not static as is the case of the current set of tools. So not only can the user select a portion of the strategy to view at a time, but that portion is rendered in the best way possible for that portion, disregarding the rest of the strategy which is not being displayed. In particular, this means that the segments closest to the top of the branch being displayed can be displayed with sufficient detail and reasonably close to each other.

***Design Principle 3: Fit as much information on the computer screen as possible, but suppress any information that only causes clutter in the display.***

When displaying strategies, this design principle can be restated as "Show more information for segments nearer the top of the strategy of the strategy than for segments far from the top of the strategy of the strategy."

The traditional approach to strategy display shows every segment in the strategy with the same amount of detail. If one zooms in, every segment gets bigger and if one zooms out every segment gets smaller. Contrast this with the fish-eye approach (see Furnas, *supra*.) that defines a single segment as the focal point of the display and displays segments with less detail the farther away they are from the focal point. This makes for a very efficient use of



screen real estate and dispenses with the need to provide a traditional zoom capability.

The focus segment is closer to the top of the strategy than its descendant segments are. Much information can be displayed about the focus segment because there is always only one focus node being displayed at any time. Much less information can be displayed about each of the descendant segments of the focus segment especially those several levels away from the focus segment, because there may be dozens, hundreds, or even thousands of those. Trying to display the same amount of information for each of these segments only makes the display so cluttered to become unusable. Information about each of these segments is suppressed when its display leads to too much clutter.

The invention herein treats the focus node specially. It's sibling segments are displayed without any of their descendant segments. The invention herein also uses a sophisticated scheme to determine when information about descendant segments is suppressed from the display based on how closely the descendant segments are rendered. When there are many descendant segments in a level, the labels for each segment's thresholds are suppressed, but the segment's split variable continues to be displayed. Where there are even more descendant segments in a level, the label for the segment's split variable is also suppressed. With additional descendant segments, the shape representing the segment in the display is drawn more compactly to permit more segments being rendered side-by-side.

***Design Principle 4: Always show the condition path.***

As described above the condition path, i.e. the path of segments from the top of the strategy of the strategy to the focus segment, is a critical piece of information to the viewer of a strategy in identifying which branch of the strategy is currently displayed and where the branch is located within the strategy. The technique herein disclosed always displays the set of conditions needed to reach the single segment currently selected as the focus segment.

***Design Principle 5: Rather than navigating by scrolling and zooming a canvas upon which the entire strategy is displayed, navigate by hopping from segment to segment.***

Scroll bars are fine for scrolling around an image, where the width of an atomic piece of information is a pixel. In the context of a strategy, the atomic piece of information is a segment, and providing scrollbars that allow one to scroll a pixel at a time is inefficient. Instead of scroll bars, navigation herein is accomplished by hopping from segment to segment. A hop is from the focus segment to any of its relatives, making that relative the new focus segment. Relative segments are sibling segments, descendant segments, or ancestor segments in the condition path.

***Design Principle 6: Provide navigational shortcuts, making it quick and easy to navigate from a segment to a neighboring segment, such as a sibling or descendant segment.***

In the traditional tools, the only way to navigate to a particular point in the strategy is via scrolling. In the herein disclosed technique, navigation occurs by hopping from the focus segment to another segment, making that other segment to the new focus segment.

5

Thus:

- Clicking any segment in the display makes that segment the focus segment.
- Clicking a condition in the condition path makes the segment corresponding to that condition the focus segment.
- Clicking a previous segment button makes the focus segment's previous sibling the new focus segment. If there is no previous sibling, the previous segment in the same level of the strategy becomes the new focus segment.
- Clicking a next segment button makes the focus segment's next sibling the new focus segment. If there is no next sibling, the next segment in the same level of the strategy becomes the new focus segment.
- Clicking an up level button makes the segment containing the focus segment become the new focus segment.

***Design Principle 7: Provide navigational cues when hopping from one segment to another.***

- 25 Because the gradual scroll provided by the scroll bar is not present, it is necessary to use another way to provide a gentle transition when the focus

segment is changed. Smooth, double-buffered animation transitions are used for this purpose.

An example of this animation is when from hopping from the focus segment to one of the focus segment's descendants. A number of the segments visible on the display initially continue to be displayed after the descendant becomes the new focus segment, but they are located in different positions in the display. The animation shows each of these segments starting in their original position and gradually migrating to their new position.

***Design Principle 8:. Maintain a consistent top of the strategy-to-leaves orientation.***

Some approaches to strategy visualization (see, for example, Lamping, et al. 1995) Lamping, J., Rao, R., and Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for viewing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, ACM, the top of the strategy of the strategy is placed dynamically anywhere in the display. The orientation from a node to its descendant nodes can vary its direction. Such approach, while useful from a purely quantitative display point-of-view, is not useful to the strategy consultant, who is used to viewing strategies that have, for example, the top of the strategy placed in a predictable, consistent location. As such, the herein disclosed approach is configured to adhere to this constraint, *i.e.* the focus segment s always at the

center, left most portion of the display. The descendant segments of the focus segment always appear to the right of the focus segment.

***Design Principle 9: Fit the strategy display into a rectangular view.***

Some approaches to strategy visualization (see, for example Lamping, *et al.* 1995) listed above require that the strategy be rendered in a circular or oval display. Computer screens are rectangular, and such an approach results in wasted screen real estate. As such, the preferred embodiment of the invention renders the display so that it makes most efficient use of a rectangular view.

***Design Principle 10: Be capable of rendering all strategies.***

Some of the existing strategy display tools are designed to render only certain, limited, kinds of strategies. For example, there are commercially available tools that include a strategy viewer which renders strategies that use the same split variable in each split on a level. The technique herein makes no assumptions about the form of the strategies that it is rendering. Every strategy that a user or software provider creates can be viewed with the invention.

***Illustrations***

Figures 1 and 2 compare the display of a portion of a strategy in a conventional strategy viewer (Figure 1) with those provided in the herein

disclosed invention (Figure 2). The strategy is too large to be displayed in full with either strategy viewer. The figures are the same size and use the same font size for node labels so as to present as fair of a comparison as possible. The portion of the strategy shown in both displays begins with a split on

5 “MonthsOnBooks.”

Here are some differences:

- The only indication given by the conventional strategy viewer of the current location in the strategy is the position of the scroll bars, which indicate that we are somewhere in the middle of the strategy. The invention not only indicates that we are one level from the top of the strategy, it also indicates that the condition that leads us to our current position is “ReissueReviewPoint in [2,4).” This means that ReissueReviewPoint must be between 2 and 4 to get to this branch.
- In the conventional strategy viewer, one cannot see the “MonthsOnBooks” split in its entirety. The “MonthsOnBooks >= 24” condition is scrolled below the window. With large strategys, this is quite a problem. Often one cannot see more than one condition of a split at any time. If one tries to zoom out enough to see all the conditions, the condition labels become too small to read. In the invention herein disclosed, this is never a problem. The left most conditions are always displayed in full as shown in this example.

- In the conventional strategy viewer, one cannot see the siblings of the “MonthsOnBooks” split. In the herein disclosed invention, the siblings are displayed above and below the “MonthsOnBooks” split. This makes for easier comparing, contrasting, and navigation between siblings.

5

- In the conventional strategy viewer, there is no clue as to what the branch looks like off the right edge of the window. One cannot tell how many levels deep it is or how many leaves it has. In the invention herein disclosed, a shrunk version of this portion of the branch is displayed to present this valuable contextual information.

10

Suppose that one wants to navigate to the parent segment of the “MonthsOnBooks” split so one can edit this split. In the invention herein disclosed, one simply clicks the left arrow button to “hop” to it. If one wants to navigate to a sibling sibling, one clicks the up arrow button or the down arrow button. Navigation is simple because one can hop from node to node. In the conventional strategy viewer, one must use the scroll bars to reposition one’s view of the canvas. This means that one has to locate (or remember) where on the canvas the node for which one is looking is displayed. For large strategies, this is like finding a needle in a haystack.

15

20

## Implementation

The presently preferred embodiment of the invention is implemented in Java.

25 The choice of Java as a development language was motivated as follows:

• **Speed:** It was important to implement the invention quickly. Industry studies indicate that with Java programmer productivity is enhanced by at least a factor of five over languages such as C++. This was borne out in the inventor's experience with the herein disclosed invention. The richness of the Java platform, such as automatic double-buffering for animations and automatic anti aliasing for smoother-looking graphics, made it possible to build the software in which the invention is implemented very rapidly. The invention, in its current state, was implemented by a single developer (a complete newcomer to Java) working half time for less than two months. The inventor's are of the opinion that even a deeply experienced C++ developer could not deliver the same level of functionality in so robust a package in such a timeframe.

• **Small Footprint:** It was important that the size of the implementation of the invention be small. The finished code is about 100Kb. It is unlikely that similar functionality could be provided in such a small package using C++.

• **Portability:** It was important that the invention herein disclosed be platform and browser neutral. Java is the natural choice.

• **Connectivity:** It was important to provide easy connectivity to servers. Once again, the built-in Java networking classes made this development straightforward.



While Java is preferred for the presently preferred embodiment of the invention, it is possible to develop the invention with other programming languages, such as in C++. However, the resulting code would be larger, more difficult to implement, and (potentially) less robust. The preferred embodiment of the herein disclosed invention has a small memory footprint and garbage collection is unlikely to pose a problem. Even if it did, the problem could be solved in a straightforward way via a simple object-pooling scheme. The herein disclosed invention is probably a poor choice for delivery via a Windows Terminal Server mechanism. The animation that is used to provide the critical navigational cues may display poorly in such an environment. Rather, the invention herein disclosed is preferably run on the client machine. Fortunately, the small size of the preferred implementation of the invention makes this a trivial issue.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Accordingly, the invention should only be limited by the Claims included below.